IFI TECHNICAL REPORTS

Institute of Computer Science,
Clausthal University of Technology

IfI-05-02

Clausthal-Zellerfeld 2005

The relationship between reasoning about privacy and default logics*

Jürgen Dix Wolfgang Faber V.S. Subrahmanian

Clausthal University of Technology, Institut für Informatik
Julius-Albert-Str. 4, 58678 Clausthal, Germany, dix@tu-clausthal.de
Dept. of Mathematics, University of Calabria
87030 Rende (CS), Italy, wf@wfaber.com
Department of Computer Science, University of Maryland
College Park, MD 20742, vs@cs.umd.edu

Abstract

There is now an incredible wealth of data about individuals, businesses and organizations. This data is freely available over the Internet to almost anyone willing to pay for it, independently of whether they are identity thieves or credit card scam artists or legitimate users. This has led to a growing need for privacy. In this paper, we first present a simple logical model of privacy. We then show that the problem of privacy may be reduced to that of brave reasoning in default logic theories, thus reducing this important problem to a well understood reasoning paradigm. By leveraging this reduction, we are able to develop efficient privacy preservation algorithms

1 Introduction

The privacy of individuals is under attack as never before. In the wake of recent terrorist events, various government agencies worldwide are seeking to acquire all kinds of private information about individuals in an effort to preserve national security. Another area where potential privacy disasters loom is in the area of medical data - many hospitals post some seemingly innocuous data on web sites (e.g. about births) but it is often possible to infer private health information about individuals. A third need for privacy mechanisms is because of poor access control and network security mechanisms that may allow outsiders to get into supposedly secure networks. In this case, there is a need to maintain privacy of data even from insiders (both genuine insiders and hackers).

*This work was supported by the European Commission under projects IST-2002-33570 INFOMIX, IST-2001-32429 ICONS, and FET-2001-37004 WASP.

Most databases have abysmal privacy mechanisms - these mechanisms by and large boil down to saying certain columns of the database are hidden from certain types of users. However, the reality of life is that many users can infer information designated private by asking queries that do not involve private information and then making common sense inferences from the answers to infer private information.

Research on privacy is now rapidly increasing because of the counterterrorism initiatives as well as because of the increasing availability of financial and health data on the web.

The primary goal of this paper is to show that there is a close connection between the problem of providing privacy preserving answers to queries and the problem of computing extensions of certain kinds of default theories. In particular, we define a linear time and linear space transformation trans of the privacy preservation problem to the problem of computing extensions of default logic theories. We prove that there is a one one correspondence between privacy preserving answers and the extensions of the default logic theory (restricted to the query) obtained by translating the privacy preservation problem into default logic via trans. Leveraging this translation, we are able to derive a suite of results on the complexity of maintaining privacy. Finally, we present an algorithm to check for privacy.

2 The Privacy Preservation Problem

In this section, we provide a simple formulation of the privacy preservation problem (**P3** for short).

We start by assuming the existence of some finite set $\mathcal{U}$ of users. Each member of $\mathcal{U}$ is a string denoting a userid.

We assume the existence of some finite set of constant symbols, function symbols and predicate symbols. As usual, a term is inductively defined as follows: (i) Each constant is a term, (ii) Each variable is a term, and (iii) if f is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term. A *ground term* is any term that contains no variable symbols. Similarly, if p is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms, then $p(t_1, \ldots, t_n)$ is an atom. A *ground atom* is any atom that contains no variable symbols. A well formed formula (wff) is inductively defined as follows. (i) Every atom is a wff, (ii) If F, G are wffs then so are $(F \wedge G), (F \vee G)$ and $\neg F$. We use WFF to denote the set of all well formed formulas in our language.

Definition 1 (logic database) *A* logic database LDB *is a finite set of ground atoms.*

Note that any standard relational database can be viewed as a logic database - if tuple $\vec{\mathsf{t}}$ is a tuple in a relation r, then $r(\vec{\mathsf{t}})$ is a ground atom.

Example 1 *We may have a small medical database containing information about the symptoms and diseases that a person p may have. Such a database may contain two predicates* symptom *and* disease*. The database may contain the following facts:*

$\mathsf{symptom}(john, s_1)$	$\mathsf{disease}(jane, aids)$
$\mathsf{symptom}(john, s_2)$	$\mathsf{disease}(john, cancer)$
$\mathsf{symptom}(john, s_3)$	$\mathsf{disease}(ed, polio)$
$\mathsf{symptom}(jane, s_1)$	
$\mathsf{symptom}(jane, s_4)$	

This little database, which we will call MedDB *will be used as a motivating example in this paper.*

The database may contain information about various individuals, businesses and organizations. These entities may wish to designate some (or all) of this information as private. For example, John and Jane may want their diseases kept private.

In addition, at any given instance t in time, each user u has some set of *background knowledge*. This background knowledge may be elicited in many ways - one such source is the set of all information disclosed to the user by the system. For example, a hospital accountant may not be allowed to see patient diagnoses, though he may see billing information about them.

Definition 2 (user model) *We assume the existence of a family of functions* $\mathsf{BK}^t : \mathcal{U} \rightarrow 2^{\mathsf{WFF}}$ *for each* t *in time, and a function* $\mathsf{Priv} : \mathcal{U} \rightarrow 2^{\mathsf{WFF}}$.

As usual, 2^X is used here to denote the power set of some set X.

Intuitively, $\mathsf{BK}^t(u)$ denotes the background knowledge of user u (which we assume to be consistent) at time t, while $\mathsf{Priv}(u)$ is the set of all formulas that the user wants to keep secret. Note that $\mathsf{BK}^t(u)$ varies as t varies. For example, as the database discloses answers to the user u, his background knowledge may increase. *Throughout the rest of this paper, we will assume that* t *is arbitrary but fixed and we address the problem of preserving privacy at time* t. As a consequence, we will usually write $\mathsf{BK}(u)$ and drop the superscript t.

Example 2 *Returning to the case of* MedDB*, John may want to keep the atom* $\mathsf{disease}(john, cancer)$ *private, while Jane may want to keep the atom* $\mathsf{disease}(jane, aids)$ *private. In this case,* $\mathsf{Priv}(john) = \{\mathsf{disease}(john, cancer)\}$*, while* $\mathsf{Priv}(jane) = \{\mathsf{disease}(jane, aids)\}$.

Likewise, consider the user acct *(denoting the accountant). This person may have the following background knowledge.*
$\mathsf{symptom}(X, s_1)$ & $\mathsf{symptom}(X, s_4) \rightarrow \mathsf{disease}(X, aids)$
$\mathsf{symptom}(X, s_2)$ & $\mathsf{symptom}(X, s_3) \rightarrow \mathsf{disease}(X, cancer)$

Definition 3 (query) *If* $A_1, \ldots, A_n$ *are all atoms, then* $(\exists)(A_1 \wedge \cdots \wedge A_n)$ *is a query.*

For example, $\mathsf{disease}(john, D)$ is a query asking what disease D John has.

Definition 4 (answer) *The* answer, $\mathsf{ANS}(Q)$*, to query* Q *w.r.t. a logic database* LDB *is the set* $\{Q\theta \mid Q\theta$ *is ground and* $\mathsf{LDB} \models Q\theta\}$ *where, as usual, the symbol "*$\models$*" denotes logical consequence.*

Example 3 *Returning to our* MedDB *example, the answer to the query* $\mathsf{disease}(john, X)$ *is the set* $\{\mathsf{disease}(john, aids)\}$. *Likewise, the answer to the query* $\mathsf{symptom}(john, X)$ *is the set* $\{\mathsf{symptom}(john, s_1),\ \mathsf{symptom}(john, s_2),\ \mathsf{symptom}(john, s_3)\}$.

In our example, we considered the case when John and Jane want their diseases kept private. However, the accountant can violate John's privacy by asking the query $\mathsf{symptom}(john, X)$. The answer she would get back is $\{\mathsf{symptom}(john, s_1),\ \mathsf{symptom}(john, s_2),\ \mathsf{symptom}(john, s_3)\}$. However, recall that the accountant has some background information - this background information includes the rule $\mathsf{symptom}(X, s_2)\ \&\ \mathsf{symptom}(X, s_3) \rightarrow \mathsf{disease}(X, cancer)$. Using this rule and the answer to her query above, the accountant can easily infer that John has cancer. The notion of a privacy preserving answer given below is intended to avoid such situations.

Definition 5 (privacy preserving answer) *Suppose* LDB *is a logic database,* $\mathcal{U}$ *is a set of users,* $u_0 \in \mathcal{U}$*, and suppose the functions* BK *and* Priv *are specified. Suppose* Q *is a query. A set* $X \subseteq \mathsf{WFF}$ *is a* privacy preserving answer *w.r.t.* $(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$ *iff:*

1. $X \subseteq \mathsf{ANS}(Q)$ *and*
2. *For all* $u \in \mathcal{U} - \{u_0\}$ *and for all* $p \in \mathsf{Priv}(u)$*, if* $\mathsf{BK}(u_0) \not\models p$ *then* $X \cup \mathsf{BK}(u_0) \not\models p$ *and*
3. *There is no* X' *such that* $X \subset X'$ *satisfies the previous two conditions.*

Intuitively, a privacy preserving answer to a query posed by user u_0 is a subset of the actual answer to the query that does not allow him to use his background knowledge to infer any new private information about any other user. Note that when user u_0 poses a query, we are only interested in preserving private information about other users u - clearly, the user u_0 can know private information about himself, as he, presumably, is the one who decides what information about him is to be kept private.

Example 4 *Let us return to the* MedDB *example, and consider the case of the obnoxious accountant. If the system knows that she has the background knowledge listed earlier, when she asks the query* $\mathsf{symptom}(john, X)$*, then it could return either of the following privacy preserving answers.*

$$\begin{aligned} Ans1 &= \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2)\} \\ Ans2 &= \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_3)\} \end{aligned}$$

Either of these two answers returns as much of the real answer as possible without making it possible for the user to infer that John has cancer.

Let us suppose that the above query was asked at time t. In this case, the accountant's background knowledge at time $t+1$ should be updated so that it includes all his previous background knowledge, plus the additional knowledge that John has symptoms s_1 and s_3. Thus, $\mathsf{BK}^t(\mathsf{acct}) = \mathsf{BK}^{t+1}(\mathsf{acct}) \cup \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2)\}$ (assuming the answer returned by the system in response to John's query at time t is $Ans1$ above. *For the sake of simplicity, throughout the rest of this paper, we assume that t is arbitrary but fixed, and that we are only interested in preserving privacy at time t.*

Example 5 *Suppose now that the system somehow knows that the accountant already had* $\mathsf{disease}(john, cancer)$ *in his background knowledge at time t (e.g. the system might know this because a doctor included the accountant on a list of people notified about John's health). In this case, revealing the entire answer* $\{\mathsf{symptom}(john, s_1)$, $\mathsf{symptom}(john, s_2)$, $\mathsf{symptom}(john, s_3)\}$ *to the query* $\mathsf{symptom}(john, X)$ *to the accountant would not violate John's privacy as the answer does not allow the accountant to infer any private facts that he did not already know. As a consequence, were the accountant's background knowledge to include the rules mentioned earlier and the additional fact* $\mathsf{disease}(john, cancer)$*, then there is only one privacy preserving answer, viz.* $\{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2), \mathsf{symptom}(john, s_3)\}$.

We emphasize that the above definition allows the background knowledge to contain some private information. A simpler definition would be to drop the *"if* $\mathsf{BK}(u_0) \not\models p$ *then"* part in (2) above. But then there would be no privacy preserving answers at all if $\mathsf{BK}(u_0)$ contained some private information. We are now ready to state the Privacy Preservation Problem (**P3**).

Problem 1 (P3$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$**)** *Suppose* LDB *is a logic database, $\mathcal{U}$ is a finite set of users,* BK *is a background knowledge function,* Priv *is a privacy function, u_0 is a user in $\mathcal{U}$ who is posing query Q to the logic database* LDB*. The* privacy preservation problem *is to find a privacy-preserving answer w.r.t.* $(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$.

The following proposition says that there is always a privacy preserving answer.

Proposition 1 *Every privacy preservation problem* ***P3***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$ *has at least one privacy preservation answer.*

A database system that seeks to preserve privacy can use the following algorithm to answer queries posed by user u_0.

algorithm PrivAns(**P3**$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$)

1. Find a privacy preserving answer Ans to query Q w.r.t. $(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$.
2. Update $\mathsf{BK}(u_0) = \mathsf{BK}(u_0), \cup Ans$.
3. Return Ans to user u_0 and halt.

The key step in this algorithm is step (1). The rest of this paper develops methods to implement step (1).

3 Translating the PP Problem to Default Logic

In this section, we provide a translation trans which takes as input, a privacy preservation problem **P3**$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$, and returns as output, a default logic theory $\Delta = (D, W)$ such that there is a one one correspondence between the solutions to the privacy preservation problem and the extensions of the default theory (restricted to the query) returned by the translation [Cadoli et al., 1997]. The consequence of this translation is that standard (and well studied) methods to evaluate default logic theories may be used to preserve privacy effectively, efficiently, and elegantly.

Definition 6 (translation trans) *Let* ***P3***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$ *be a privacy preservation problem. The* translation*,* ***trans***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$ *of a privacy preservation problem into default logic is the default logic theory* $\Delta = (D, W)$ *where:*

$$
\begin{aligned}
W &= \mathsf{BK}(u_0). \\
D &= \{\frac{: f}{f} \mid f \in \mathsf{LDB}\} \bigcup \\
&\quad \{\frac{p :}{\neg p} \mid (\exists u \in \mathcal{U} - \{u_0\})\ p \in \mathsf{Priv}(u) \\
&\quad\ \ \textit{and}\ \ \mathsf{BK}(u_0) \not\models p\}.
\end{aligned}
$$

We now present an example to show what the result of transforming the privacy preservation problem into default logic looks like.

Example 6 *Let us return to the case of the accountant. In this case,* W *consists of the two rules* $\begin{array}{l}\mathsf{symptom}(X, s_1) \,\&\, \mathsf{symptom}(X, s_4) \rightarrow \mathsf{disease}(X, aids) \\ \mathsf{symptom}(X, s_2) \,\&\, \mathsf{symptom}(X, s_3) \rightarrow \mathsf{disease}(X, cancer).\end{array}$ *In addition,* D *consists of the following defaults:*

$$
\frac{: \mathsf{symptom}(john,s_1)}{\mathsf{symptom}(john,s_1)} \quad
\frac{: \mathsf{symptom}(john,s_2)}{\mathsf{symptom}(john,s_2)} \quad
\frac{: \mathsf{symptom}(john,s_3)}{\mathsf{symptom}(john,s_3)} \quad
\frac{: \mathsf{symptom}(jane,s_1)}{\mathsf{symptom}(jane,s_1)} \quad
\frac{: \mathsf{symptom}(jane,s_4)}{\mathsf{symptom}(jane,s_4)}
$$

$$
\frac{: \mathsf{disease}(jane,aids)}{\mathsf{disease}(jane,aids)} \quad
\frac{: \mathsf{disease}(john,cancer)}{\mathsf{disease}(john,cancer)} \quad
\frac{: \mathsf{disease}(ed,polio)}{\mathsf{disease}(ed,polio)}
$$

$$
\frac{\mathsf{disease}(jane,aids) :}{\neg\mathsf{disease}(jane,aids)} \quad
\frac{\mathsf{disease}(john,cancer) :}{\neg\mathsf{disease}(john,cancer)}
$$

Note that we are assuming here that Ed has not marked his disease as being a private fact.

Note that this translation uses linear space. The time complexity of the translation depends on the complexity of checking entailment. For example, assuming a finite number of constants in our language (reasonable) and assuming that all rules in BK are definite

clauses, then the translation is implementable in polynomial time. But if BK can consist of arbitrary first order formulas, then the translation can take exponential time.

Before presenting our central theorem linking privacy preserving answers and extensions of default theories, we remind the reader of some basic terminology associated with default theories. Given a default $d = \frac{\alpha:\beta}{\gamma}$, we use the notation $pre(d)$ to denote α, $j(d)$ to denote β and $c(d)$ to denote γ. In addition, given any default theory $\Delta = (D, W)$, we may associate with Δ, a mapping Γ_Δ which maps sets of wffs to sets of wffs. $\Gamma_\Delta(Y) = \mathsf{CN}(W \cup \{pre(d) \rightarrow c(d) \mid j(d) \text{ is consistent with } Y\}$. As usual, the function $\mathsf{CN}(X)$ denotes the set of all first order logical consequences of X. A set Y of wffs is an *extension* of Δ iff $Y = \Gamma_\Delta(Y)$.

We are now ready to present a key result linking the privacy preservation problem and default logic extensions. Suppose we consider any privacy preservation problem. The privacy preserving answers to that privacy preservation problem are in a one-one correspondence with the consistent extensions of the translation (restricted to the query) of the privacy preservation problem into default logic (using the translation trans shown in Definition 6).

Theorem 7 *Suppose that A is an atom and that **P3**$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, A)$ is a privacy preservation problem and $\mathsf{trans}(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, A) = \Delta = (D, W)$. Then: X is a solution to the above privacy preservation problem iff there is a consistent extension E of $\Delta = (D, W)$ such that $X = \{A\theta \mid A\theta \in E \cap \mathsf{LDB}\}$.*

Proof. We first formulate a useful abstract lemma.

Lemma 8 *Let W, LDB and P be consistent sets of formulae s.t. $W \cup LDB$ is consistent as well. Let $D_P = \{\frac{p\ :}{\neg p}\ :\ p \in P\}$ and $D_{LDB} = \{\frac{:\ f}{f}\ :\ f \in LDB\}$.*

Then the consistent extensions of the theory $(D_P \cup D_{LDB}, W)$ are the sets $Cn(W \cup \{f :\ f \in F\})$ where F is a subset of LDB that is maximal wrt. set inclusion (i.e. there is no larger set F' such that $W \cup \{f :\ f \in F'\} \not\models p$ for all $p \in P$).

The proof of our main theorem is an application of our lemma. Suppose X is a solution to **P3**$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, A)$ and let $\mathsf{trans}(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, A) = \Delta = (D, W)$. Then we let $F := X$, $W := \mathsf{BK}(u_0)$ and $P := \{p :\ (\exists u \in \mathcal{U} - \{u_0\})\ p \in \mathsf{Priv}(u) \text{ and } \mathsf{BK}(u_0) \not\models p\}$ and apply our lemma. The set $Cn(W \cup \{f :\ f \in F\})$ is an extension (it is maximal because of (3) and (2) in the definition of a privacy preserving answer).

Conversely let be given a consistent extension E of $\mathsf{trans}(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, A)$ and consider $X := \{A\theta \mid A\theta \in E \cap \mathsf{LDB}\}$. Our lemma implies that X is a subset of LDB that is maximal. Therefore X is also a privacy preserving answer (if there were a larger X' satisfying (2) in the definition of pp answer, then E would not be maximal and thus not be an extension).

Proof of the lemma. Clearly the sets $Cn(W \cup \{f :\ f \in F\})$ where F is a maximal subset of LDB are extensions of the default theory: the defaults in D_P do not apply

and we are left with a supernormal default theory (the result follows from well-known characterizations in default logic, see eg. [Dix, 1992, Marek and Truszczyński, 1993]).

Conversely, let E be a consistent extension. Then no default in D_P applies. Because extensions are grounded and we are dealing with a supernormal theory, E must have the form $Cn(W \cup \{f : \ f \in F\})$ for a subset F of LDB. Because E is maximal (no other extension can contain E), the set $Cn(W \cup \{f : \ f \in F\})$ is maximal in the sense defined in the lemma.

The preceding theorem applies to *atomic* queries. A straightforward extension of the above proof gives us the following theorem which applies to arbitrary queries.

Corollary 1 *Suppose that* ***P3***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$ *is a privacy preservation problem and that* ***trans***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q) = \Delta = (D, W)$*. Then:* X *is a solution to the above privacy preservation problem iff there is a consistent extension* E *of* $\Delta = (D, W)$ *such that* $X = \{Q\theta \mid Q\theta \in E \cap \mathsf{LDB}\}$.

In order to illustrate this theorem, we revisit the example privacy preservation problem and its default logic translation that we presented earlier.

Example 9 *Let us return to the* MedDB *example. Consider the privacy preservation problem of Example 4 and the default logic translation shown in Figure 6. As seen in Example 4, there are two privacy preserving answers to this problem. They are:*

$$\begin{aligned} Ans1 &= \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2)\} \\ Ans2 &= \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_3)\}. \end{aligned}$$

The default logic translation of this privacy preservation problem shown in Example 6 has exactly four consistent extensions $E_1, \ldots, E_4$.

$$\begin{aligned} E_1 &= \mathsf{CN}(W \cup \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2), \\ &\quad \mathsf{symptom}(jane, s_1), \mathsf{disease}(ed, polio)\}). \\ E_2 &= \mathsf{CN}(W \cup \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_3), \\ &\quad \mathsf{symptom}(jane, s_1), \mathsf{disease}(ed, polio)\}). \\ E_3 &= \mathsf{CN}(W \cup \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2), \\ &\quad \mathsf{symptom}(jane, s_4), \mathsf{disease}(ed, polio)\}). \\ E_4 &= \mathsf{CN}(W \cup \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_3), \\ &\quad \mathsf{symptom}(jane, s_4), \mathsf{disease}(ed, polio)\}). \end{aligned}$$

However, if we restrict our interest to answers to the query $\mathsf{symptom}(john, X)$ *in the above extensions, then extensions* E_1, E_4 *only contain* $\{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2)\}$ *while* E_2, E_3 *only contain* $\{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_3)\}$. *These restrictions of the extensions are in a one-one correspondence with the privacy preserving answers to the query posed by the accountant.*

4 Complexity of Privacy Preservation

In this section, we analyze the complexity of the privacy preservation problem.

Computing a privacy-preserving answer typically involves "guessing" a subset of answers, and subsequently checking it with respect to privacy preservation and maximality. Intuitively, this computational task has a correspondence to common non-monotonic reasoning tasks, because the maximality condition for privacy-preserving answers has its counterpart as minimality conditions in non-monotonic semantics, while guessing a model candidate and checking it on a set of formulae is even more closely related.

It therefore does not come as a surprise that a non-monotonic logic allows for an apt representation of the privacy preservation problem. Concerning the complexity analysis, we can indeed leverage the translation trans to use well-known results concerning the complexity of default logic in order to prove membership of various subclasses of **P3**.

As already shown in [Reiter, 1980], default reasoning involving function symbols is undecidable. Note that Definitions 5 and 6 involve checking $\mathsf{BK}(u_0) \not\models p$, which is clearly undecidable for arbitrary first-order formulae. We will therefore focus on decidable fragments. In particular, we will assume in our analyses below that problems are restricted to those for which deciding $\mathsf{BK} \not\models p$, $p \in \mathsf{Priv}$ is feasible in polynomial time. We will focus on theories in a Datalog setting, the data complexity of which corresponds to propositional default theories.

Then, membership can be seen by virtue of trans and the shape of formulae in BK and Priv. In particular, brave reasoning for non-disjunctive default theories is NP-complete (see e.g. [Kautz and Selman, 1991, Stillman, 1990] for such classes), while brave reasoning for arbitrary default theories is Σ_2^P-complete, see [Gottlob, 1992] and [Stillman, 1992].

We thus consider **P3**s with the following restrictions:

1. We vary $\mathsf{BK}(u)$ to be an arbitrary theory, a non-disjunctive theory, and a set of facts.
2. We vary $\mathsf{Priv}(u)$ to be a set of arbitrary formulas, a non-disjunctive theory, and a set of facts.

Table 1 summarizes our results on the complexity of privacy preservation in the Datalog case.

Theorem 10 *The data complexity for **P3** problems without function symbols under various syntactic restrictions are as reported in Table 1. Completeness holds for* NP *and* Σ_2^P *results.*

Next, we will prove some of the hardness results.

Corollary 2 ***P3*** *with* BK *containing non-disjunctive rules and* Priv *made of facts is hard for* NP.

Priv/BK	Facts	Non-disjunctive	Arbitrary
Facts	P	P	Σ_2^P
Non-disjunctive	NP	NP	Σ_2^P
Arbitrary	Σ_2^P	Σ_2^P	Σ_2^P

Table 1: Data Complexity of Privacy Preservation Problems

Proof. We show NP-hardness by a reduction from 3SAT to a **P3** in which BK contains only rules with negation on LDB predicates and in which Priv contains only one fact: Given a CNF $\phi = \bigwedge_{i=1}^{n} L_{i,1} \vee L_{i,2} \vee L_{i,3}$, we create a **P3** with $\mathsf{LDB} = \{c_i \mid c_i \text{ is an atom in } \phi\} \cup \{q\}$, two users u_0, u_1, $\mathsf{BK}(u_0) = \{L'_{i,1} \wedge L'_{i,2} \wedge L'_{i,3} \rightarrow unsat\}$, where $(\neg x)' = x$ and $x' = \neg x$. Finally, $\mathsf{Priv}(u_1) = \{unsat\}$, and $Q = q$. It is not hard to see that q is an answer iff ϕ is satisfiable: If q is an answer, then a truth assignment can be obtained from the subset $X \subseteq \mathsf{LDB}$ in which exactly the c_i in X are interpreted as true. Since $unsat$ does not hold for this X, no conjunct in ϕ evaluates to false under this assignment, which therefore satisfies ϕ. Conversely, if ϕ is satisfiable, each cardinality maximal satisfying truth assignment induces an $X \subseteq \mathsf{LDB}$, such that $X \cup BK(u_0) \not\models unsat$.

Corollary 3 ***P3*** *with empty* BK *and arbitrary* Priv *is hard for* Σ_2^P.

Proof. We show Σ_2^P-hardness by a reduction from a $QBF_{2,\exists}$ to a **P3** in which BK is empty and Priv contains arbitrary formulae. Consider $\psi = \exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m \phi$, where ϕ is a propositional formula. We create a **P3** with $\mathsf{LDB} = \{x_1, \ldots, x_n\} \cup \{q\}$, two users u_0, u_1, $\mathsf{Priv}(u_1) = \{\neg E\}$, and $Q = q$. An answer X induces a valuation ν of the existentially quantified variables. Then, no extension ν' of ν to the universally quantified variables can exist such that E is false, hence ψ is valid. Conversely, if ψ is valid, each cardinality maximal satisfying truth assignment for $x_1, \ldots, x_n$ induces an answer.

This proof can easily be adapted such that $\mathsf{BK}(u_0)$ contains the arbitrary formula $(\neg E) \rightarrow unsat$ and $\mathsf{Priv}(u_1)$ contains only $unsat$.

All complexity results above refer to propositional theories or data complexity, in our setting this means that only LDB is considered as input, while especially BK and Priv are considered to be fixed. For considering program complexity, we can adapt the data complexity results by using techniques from [Gottlob et al., 1999]. Due to space constraints, we do not present proofs.

Theorem 11 *The program complexity for* ***P3*** *problems without function symbols under various syntactic restrictions are as reported in the table below.*

Priv/BK	Facts	Non-disj.	Arbitrary
Facts	EXPTIME	EXPTIME	NEXPTIME$^{\mathrm{NP}}$
Non-disj.	NEXPTIME	NEXPTIME	NEXPTIME$^{\mathrm{NP}}$
Arbitrary	NEXPTIME$^{\mathrm{NP}}$	NEXPTIME$^{\mathrm{NP}}$	NEXPTIME$^{\mathrm{NP}}$

To summarize, the results in this section confirm that default logic is indeed a suitable choice to represent **P3**s.

5 Privacy Preservation Algorithm

In this section, we describe an algorithm to preserve privacy that leverages our translation of the privacy preservation problem to default logic. First and foremost, we recall the important observation of [Baral and Subrahmanian, 1993] that Reiter's Γ_Δ operator is anti-monotonic - hence, the operator Γ^2_Δ that applies Γ_Δ is monotonic. As a consequence, Γ^2_Δ has both a least fixpoint and a greatest fixpoint, denoted $\mathsf{lfp}(\Gamma^2_\Delta)$ and $\mathsf{gfp}(\Gamma^2_\Delta)$ respectively.

Theorem 12 ([Baral and Subrahmanian, 1993]) *Recall the following properties:*

1. *If* $Y_1 \subseteq Y_2$ *then* $\Gamma_\Delta(Y_2) \subseteq \Gamma_\Delta(Y_1)$.
2. Γ^2_Δ *has a least and a greatest fixpoint, denoted respectively as* $\mathsf{lfp}(\Gamma^2_\Delta)$ *and* $\mathsf{gfp}(\Gamma^2_\Delta)$.
3. $\Gamma_\Delta(\mathsf{lfp}(\Gamma^2_\Delta)) = \mathsf{gfp}(\Gamma^2_\Delta)$.

An immediate consequence of the above theorem is that one can compute extensions of default theories by first computing $\mathsf{lfp}(\Gamma^2_\Delta)$ and $\mathsf{gfp}(\Gamma^2_\Delta)$. Anything in $\mathsf{lfp}(\Gamma^2_\Delta)$ is true in all extensions, while anything not in $\mathsf{gfp}(\Gamma^2_\Delta)$ is false in all extensions. We can therefore start by computing both $\mathsf{lfp}(\Gamma^2_\Delta)$ and $\mathsf{gfp}(\Gamma^2_\Delta)$. If $\mathsf{lfp}(\Gamma^2_\Delta)$ is not an extension, we nondeterministically add things in $\mathsf{gfp}(\Gamma^2_\Delta)$ to the default theory and iteratively compute the least fixpoint of Γ^2_Δ w.r.t. the modified theory. This algorithm for arbitrary default theories gives rise to the following specialization for computing privacy preserving answers.

P3Alg($\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q$)
$\Delta = \mathsf{trans}(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q) = (D, W)$;
$Todo = \mathsf{LDB} \cap (\mathsf{gfp}(\Gamma^2_\Delta) \setminus \mathsf{lfp}(\Gamma^2_\Delta))$;
if $\mathsf{lfp}(\Gamma^2_\Delta) = \Gamma_\Delta(\mathsf{lfp}(\Gamma^2_\Delta))$ **then**
    $done = true$;
while $Todo \neq \emptyset \wedge \neg done$ **do**
    Nondeterministically select an $a \in Todo$;
    Let $\Delta = (D, W \cup \{a\})$;
    if $\mathsf{lfp}(\Gamma^2_\Delta) = \Gamma_\Delta(\mathsf{lfp}(\Gamma^2_\Delta))$ **then**
        $done = true$;
    else

$Todo = Todo \setminus \{a\}$;

% **end-while**

return LDB $\cap$ lfp(Γ^2_Δ);

The algorithm proceeds as follows: First the problem is translated to a default theory using trans. Subsequently, the least and greatest fixpoint of Γ^2_Δ are computed. Anything which is in the greatest, but not in the least fixpoint can or cannot be true in some extension, so we store it in $Todo$ to nondeterministically assume its truth.

The crucial point here is that we restrict these nondeterministic choices to LDB, which can dramatically decrease the search space. Then we enter the nondeterministic phase of the algorithm, in which a truth assignment for $Todo$ is generated until a fixpoint (i.e., an extension) is reached, if at all. As a final step, a projection of the extension onto LDBis generated.

The following theorem states that the above algorithm is always guaranteed to return the correct answer.

Theorem 13 *Consider a privacy preservation problem* ***P3***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$. *Then the algorithm* ***P3Alg***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$ *returns X iff X is a privacy preserving answer to* ***P3***$(\mathsf{LDB}, \mathcal{U}, \mathsf{BK}, \mathsf{Priv}, u_0, Q)$.

We have thus given an effective and also efficient (w.r.t. to general algorithms computing answers to default theories) algorithm for computing privacy preserving answers.

6 Related Work and Conclusions

Security and privacy of information are closely related. There has been extensive work on privacy and security for many years now [M. Winslett and Qian, 1994, P. Bonatti and Subrahmanian, 1995, Cuppens and Demolombe, 1997, Samarati and Sweeney, 1998]. A body of work in the field [M. Winslett and Qian, 1994, P. Bonatti and Subrahmanian, 1995] set up the security problem as that of inferring a maximal subset of the answer to a query so that no secrets are violated. Algorithms were also given to determine how to update the database so that security and privacy are preserved. Another body of work [Samarati and Sweeney, 1998] determines how to generalize answers (rather than choose a subset). Our work is related to the former category.

In contrast to the above body of work, we are aware of no works that ties well known nonmonotonic logic formalisms such as default logic to the privacy preservation problem. This paper is a first step in this regard. As shown by the P3Alg algorithm and the complexity results derived in this paper, the relationships between privacy preservation and default logics can lead to results in one domain being applicable and beneficial to another. Our future work will focus on leveraging the relationship between default logic and privacy even further so that the rich experience gained in implementing default logics can be applied fruitfully to the privacy domain.

References

[Baral and Subrahmanian, 1993] Baral, C. and Subrahmanian, V. (1993). Dualities Between Alternative Semantics for Logic Programming and Non-Monotonic Reasoning. *Journal of Automated Reasoning*, 10(3):399–420.

[Cadoli et al., 1997] Cadoli, M., Eiter, T., and Gottlob, G. (1997). Default Logic as a Query Language. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):448–463.

[Cuppens and Demolombe, 1997] Cuppens, F. and Demolombe, R. (1997). A Modal Logical Framework for Security Policies. In *Proc. ISMIS*, pages 579–589.

[Dix, 1992] Dix, J. (1992). Default Theories of Poole-Type and a Method for Constructing Cumulative Versions of Default Logic. In Neumann, B., editor, *Proc. of 10th European Conf. on Artificial Intelligence ECAI 92*, pages 289–293. John Wiley & Sons.

[Gottlob, 1992] Gottlob, G. (1992). Complexity Results for Nonmonotonic Logics. *Journal of Logic and Computation*, 2(3):397–425.

[Gottlob et al., 1999] Gottlob, G., Leone, N., and Veith, H. (1999). Succinctness as a Source of Expression Complexity. *Annals of Pure and Applied Logic*, 97(1–3):231–260.

[Kautz and Selman, 1991] Kautz, H. and Selman, B. (1991). Hard Problems for Simple Default Logics. *Artificial Intelligence*, 49:243–279.

[M. Winslett and Qian, 1994] M. Winslett, K. S. and Qian, X. (1994). Formal Query Languages for Secure Relational Databases. *ACM Transactions on Database Systems*, 19(4):626–662.

[Marek and Truszczyński, 1993] Marek, W. and Truszczyński, M. (1993). *Nonmonotonic Logics; Context-Dependent Reasoning*. Springer, Berlin, 1st edition.

[P. Bonatti and Subrahmanian, 1995] P. Bonatti, S. K. and Subrahmanian, V. (1995). Foundations of Secure Deductive Databases. *IEEE Transactions on Knowledge and Data Engineering*, 7(3):406–422.

[Reiter, 1980] Reiter, R. (1980). A Logic for Default Reasoning. *Artificial Intelligence*, 13(1–2):81–132.

[Samarati and Sweeney, 1998] Samarati, P. and Sweeney, L. (1998). Generalizing Data to Provide Anonymity when Disclosing Information. In *Proceedings ACM Symp. on Principles of Database Systems*.

[Stillman, 1990] Stillman, J. (1990). It's Not My Default: The Complexity of Membership Problems in Restricted Propositional Default Logic. In *Proceedings AAAI-90*, pages 571–579.

[Stillman, 1992] Stillman, J. (1992). The Complexity of Propositional Default Logic. In *Proceedings AAAI-92*, pages 794–799.

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Wojciech Jamroga
Contact: wjamroga@in.tu-clausthal.de
URL: http://www.in.tu-clausthal.de/~wjamroga/techreports/

The IfI Review Board